

Honours Project 2005

Representing Software Traceability using UML and XTM with an investigation into Traceability Patterns

David Hollings
University of Cape Town
Dept. of Computer Science
South Africa
dholling@cs.uct.ac.za

Kieran Sharpey - Schafer
University of Cape Town
Dept. of Computer Science
South Africa
ksharpey@cs.uct.ac.za

Justin Kelleher
University of Cape Town
Dept. of Computer Science
South Africa
jkr@cs.uct.ac.za

Abstract

Despite the importance of requirements traceability in developing software systems, there are few applications that provide useful requirements management techniques. This paper looks at XML Topic Maps as a new technique for the representation of requirements traceability in order to improve requirements management, analysis and visualisation.

The aim of this project is to investigate the validity of using XML Topic Maps to represent requirements traceability. A software application is developed and used to prove this concept.

The system and project goals are then tested. It is shown that the use of XML Topic Maps is probably an improved method for the management, analysis and visualisation of requirements traceability. Furthermore, it is concluded that an improved visualisation technique will be required for the intuitive navigation of the requirements traceability graphs.

Introduction: Traceability Problem and solution

Software development has various stages consisting of requirements elicitation and management, design, implementation and testing. Each of these stages produce artefacts: use cases, class diagrams, code modules and test cases for example. *Traceability* is the concept that these artefacts affect each other across all stages of development. For example it denotes the fact that a class in code is traced to its class diagram and a change in one will cause a change in the other. Traceability is useful as it allows project managers to see how changes, failures and performance issues will affect the rest of the project.

This paper focuses on a solution for *requirements traceability*, a concept introduced by Gotel and Finkelstein [11]. Requirements traceability ensures that each requirement is satisfied by the final system implementation. This is done by ‘tracing’ the requirements to artefacts in all stages of the project. Similarly a set of requirements can affect each other and therefore traceability

links will exist between them. It is this aspect of requirements traceability that this paper focuses.

Although traceability is agreed to be essential, it is often avoided both through unfamiliarity and reluctance to maintain documentation. Traceability links can easily be broken if changes are not propagated throughout the system and documentation is not correctly updated [3, 9]. These factors make accurate traceability hard to achieve in practice [1].

Previous attempts at providing tools to manage traceability in a project have been difficult to use or limited in functionality. This could be attributed to the lack of a tool that can visualise, manage and analyse traceability in a project. A simple new form of visualising traceability is not the answer but rather a platform or medium is needed to be able to study traceability. Such a platform would benefit contemporary research areas such as traceability visualisation, impact analysis as well as requirements management.

In this paper we present an innovative method for modelling requirements traceability using XML Topic Maps (XTM). This new data format is particularly useful as it is designed for the description of data and the relationships therein. We describe an ontology that defines this model, based on the UML description of requirements traceability introduced by Letelier [6], and the XTM syntax specified by topicmaps.org in 2001 [7]. The XTM model means that the descriptions of requirements now have a standard representation, are machine processable

and are interchangeable between programs.

In this paper we will illustrate the potential of this model by describing a system that is able to:

- Translate UML modelled requirements into XTM
- Manage a requirement's attributes and traces
- Show a visual representation of the model's requirements and traces
- Enable navigation between requirements using these traceability links
- Analyse the attributes and traces of a set of requirements

We further describe how this functionality demonstrates management, visualisation and analysis of requirements traceability.

This paper will describe the background and interoperability of the various technologies and techniques used. An overview of the implemented system will then be given followed by sections discussing testing, conclusions and future work.

Background Traceability & XTM

In the search for making traceability easier and cheaper to implement, the argument of automated tools for traceability has been raised.

Currently traceability and impact of change are monitored by manual traversal of the traces in a system. Travelling downwards on traces through the development cycle can be used to monitor the impact of change. Travelling

upwards towards requirements will help determine if the scope of the requirement is still covered [4].

Spanoudakis et al [10] make the argument that the support for traceability in existing tools is the root cause of its lack of implementation. They propose that the main problem is that no tools are able to automate the generation and maintenance of traceability links. This argument is continued by the charge that most academic research assumes that traceability links must be identified manually.

An automated, or even partially automated, solution would be attractive to large industrial development projects that need to monitor change on vast collections of data. [10]

Background XTM

Topic maps are an ISO standard first published as ISO/IEC 13250 in 2000. The ISO 13250 standard defines a basic model and a SGML-based syntax. In 2001 TopicMaps.org, was formed to define a web-enabled syntax [5] for topic maps. This syntax was included as an annex to the ISO/IEC 13250 specification as XTM 1.0 and included a syntax defined in XML as well as using URI's for linking [8].

The purpose of the Topic Map syntax is to create a superimposed layer of information metadata identifying a collection of related subjects of interest [2, 12]. This layer fundamentally consists of Topics, Occurrences and Associations all linked together in order to form an intuitive and navigable representation of underlying data [2, 5].

The data described may be stored virtually anywhere, such as a spreadsheet, database, webpage or the Topic Map itself and is pointed to by URI's within the Topic Map.

Another powerful feature of Topic Maps is the ability to merge maps together. Topic Maps may be merged if both have topics that refer to the same subject [8].

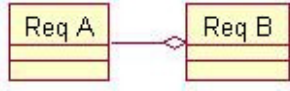
Furthermore, the topic map community is in the process of standardizing a query language [5] known as TMQL. No standard has yet been approved, but much work is being done on providing a query language based on predicate logic with a syntax similar to SQL [12].

The associative and descriptive nature of a Topic Map makes it useful to use as a stand alone data structure, which records relationships only and has no mappings to external (hyperlinked) information at all. Power [8] distinguishes between the practices of keeping data in the Topic maps structure itself and using it to reference existing data as it was intended. This allows the customization of a Topic Map solution to a particular problem, depending on the sources of information, and gives further opportunities for the uses of Topic Maps.

Despite the relatively new status of the Topic Map specification, a wide variety of commercial and open-source solutions are already being offered that make use of the power of Topic Maps. All of the implementations have certain features in common such as an API to retrieve and manipulate data. Such implementations are: TM4J, Ontopia Omnigator, Gooseworks Topic Map toolkit.

Designing the XTM Model

Table 1: Conversion from UML to Topic Map

SE term	UML	XML Topic Map
Requirement:	Class	Topic
Attribute of Requirement:	Class member	Topic occurrence of type = attribute name with value in kept in topic map and not external reference.
Functional/ Non Functional:	Class Stereotype	Topic given type corresponding to Stereotype using <code><instanceOf></code>
Association:		An association with two <code><member></code> s referencing the requirements. An <code><instanceOf></code> determines the type of the association.

The aim of the project was to find a new way of representing traceability that will increase the reasons for its use and remove the drawbacks of current techniques.

In this section we will describe how we set up the requirements model and how we gave this a description in XTM.

The first stage was to construct a UML model of the requirements set as described by Letelier [6]. He states that traceability in a project can be configured in the following stages:

1. Identify all artefacts relevant to describing the traceability of the project.
2. Define dependency relationships between artefacts.
3. Establish types of traceability links that are particular to the project.
4. Define criteria to implicitly derive the types of traceability links.

Requirements are modelled in a class diagram, have attributes (such as risk, cost and priority) and model traces as associations of a standard UML types of:

Dependency, Inheritance, Aggregation and Composition.

An example of this is shown in Figure 2:

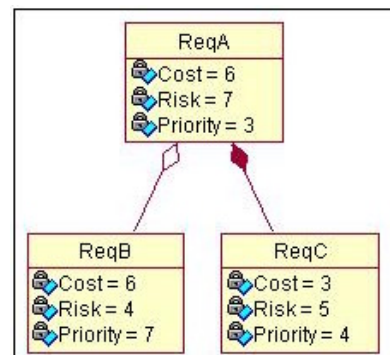


Figure 2: Requirements modelled in UML

We added to this by assigning stereotypes to each requirement class in order to define its type – functional or non-functional.

Our next stage was to create a description of this UML model in the XTM syntax.

Topics, associations and occurrences are used to represent requirements, traces and attributes respectively. Types, for requirements, traces and attributes, are applied using the `<instanceOf>` tag from the XTM syntax. This tags specific

purpose is to apply ‘type’ to topics, associations and occurrences. Table 1 illustrates the correspondence between the different models.

The XTM syntax was chosen for our new representation of requirements traceability since:

- XML Topic Map standard enforces standard structure
- Stores association types and members in association
- Stores attribute types and values
- XML syntax allows for easy data extraction
- Merging of XTM allows common requirements to be traced across sub-projects.
- XTM parsing and visualisation techniques are already available

Merging is particularly interesting in the realm of traceability as merging of traces is needed: “Fundamentally, concurrent engineering relies on the capability to merge pieces of work done in a concurrent way on the same object. Merging objects is thus a central issue. Trace information should be captured and traceability between related fragments should be re-established” [11]. Thus traceability represented in a Topic Map could be merged on common points.

We will now describe the system built to demonstrate the model.

Approach

We demonstrate the power of our XTM model by implementing the following system, illustrated in Figure 2.

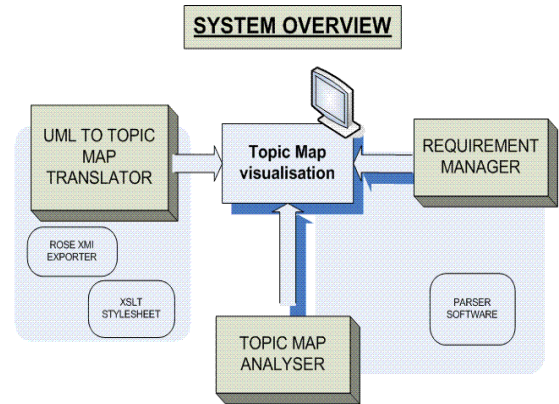


Figure 2: Example System Overview

A pre-existing Topic Map visualisation module sits at the centre of the system, namely TMNav: a product of the Topic Maps for Java (TM4J) project.

This visualisation module can be passed XTM files by a module that converts UML diagrams into our XTM syntax model. This is done via an XSLT style-sheet that converts from a Rational Rose XMI export to XTM.

The visualisation module can also be passed XTM files by a *UML to Topic Map translator* or a *Requirement Manager* module that can create requirements in a topic map format. This module is also able to view and edit topic maps that are currently in the centre module.

A 3rd module, the *topic map analyser*, is also able to view the content of the visualisation module’s topic maps for the purpose of filtering requirements from a model, based on constraints on its inherent characteristics (i.e. attributes and traces).

The Requirement Manager and the Topic Map analyser both rely on a core package that parses the XTM files and returns the information retained in easy to manage data structures. This package makes it easy for programmers to manipulate and process the data of the requirement set.

Testing and Validation

We conducted an evaluation of the system by testing three aspects of the final implementation:

1. How well the system achieved the original functional goals
2. The validity of the XML Topic Map files generated
3. The validity of using XML Topic Maps to represent requirements traceability

Functional goals and requirements were tested in order to ensure valid operation. The requirements traceability management and validation components were found to operate as required.

The validity of the topic map files created at runtime was tested by validating the XTM files against the official XTM Document Type Definition (DTD). The outputted files were realised to be valid and well formed.

Finally, we tested the concept of representing requirements traceability using XML Topic Maps by conducting extensive user testing. Users involved with this phase of testing showed a good understanding of the resultant XTM models. The users found the XTM models a more intuitive visual representation when compared against standard traceability matrices and the earlier UML models. One important realisation was, however, the need for an entire overview of the resultant topic map structure as opposed to the single requirement view provided. This lack of an intuitive overview of the entire structure resulted in various problems with the navigability of the traceability graph.

Conclusions

The representation of requirements traceability using UML and subsequently XTM was viewed as largely successful. Both models for

visualising requirements traceability allow for an intuitive method of managing and analysing traceability graphs.

We concluded that the resulting XTM models represent a more understandable representation of traceability graphs, however, a more intuitive technique for the visualisation of XTM is required.

The management and analysis of requirements traceability was successfully implemented. These implementations allow an extendable platform for a requirements management application using the XTM data format for visualisation.

Future Work

The systems functionality should further be extended to allow for a visual aid when investigating the impact of changing requirements. These changes should further be automated in order to propagate change throughout a traceability graph. This automation of change propagation could be conveniently managed by integration of a Bayesian network created for the management of requirement attributes.

Additionally, a constraint language should be investigated in order to define a specific syntax for the resulting model. This constraint language should enforce a standardised approach to visually representing requirements traceability to allow for interoperability between sub-projects.

References

- [1] Arkley, P., Mason, P., Riddle, S. 2002. Enabling Traceability. From proceedings of the First International Workshop on Traceability, Edinburgh, 28th September

- 2002, in conjunction with the 17th IEEE International Conference on Automated Software Engineering
- [2] Biezunski, M., Newcomb, S.R. 2001. XML Topic Maps: Finding Aids for the Web. From IEEE Multimedia, Vol. 2, Ch 8, pg 108 - 112. June 2001
 - [3] Cleland-Huang, J., Schmelzer, D. 2003. Dynamically Tracing Non-Functional Requirements through Design Pattern Invariants. From proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE '2003), Montreal, Canada, 10-2003.
 - [4] Dick, J. 2002. Rich Traceability. From proceedings of the First International Workshop on Traceability, Edinburgh, 28th September 2002, in conjunction with the 17th IEEE International Conference on Automated Software Engineering.
 - [5] Lacher, M.S., Decker, S. 2001. On the Integration of Topic Maps and RDF Data, from Proceedings of Extreme Markup Languages 2001. Montréal, Quebec, August, 2001.
 - [6] Letelier, P. 2002. A Framework for Requirements Traceability in UML-based Projects. From proceedings of the First International Workshop on Traceability, Edinburgh, 28th September 2002, in conjunction with the 17th IEEE International Conference on Automated Software Engineering.
 - [7] Pepper, S., Moore, G. 2001. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification. Aug. TopicMaps.org Consortium, Web site: www.topicmaps.org.
 - [8] Power, R. 2003. Topic Maps for Context Management, from proceedings of the 1st international symposium on Information and communication technologies. Vol. 49, Dublin, Ireland, July, 2003, pp 199-204.
 - [9] [46] Richardson, J., Green, J. 2003. Traceability Through Automatic Program Generation. From proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE '2003), Montreal, Canada, 10-2003.
 - [10] Spanoudakis, G., Zisman, A. Perez-Minana, E., Krause, P. 2004. Rule-based generation of requirements traceability relations. Journal of Systems and Software, 2004
 - [11] Strasunskas, D. 2002. Traceability in Collaborative Systems Development from Lifecycle Perspective. From proceedings of the First International Workshop on Traceability, Edinburgh, 28th September 2002, in conjunction with the 17th IEEE International Conference on Automated Software Engineering.
 - [12] Seedorf, S., Korthaus, A., Aleksy, M. 2005. Creating a topic map query tool for mobile devices using J2ME and XML. In Proceedings of the 4th international Symposium on information and Communication Technologies (Cape Town, South Africa, January 03 - 06, 2005). ACM International Conference Proceeding Series, vol. 92. Trinity College Dublin, 111-116.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.